

```

LLL                      0000000000    GGGGGGGGGGGG    IIIIIIIII    NNN    NNN
LLL                      0000000000    GGGGGGGGGGGG    IIIIIIIII    NNN    NNN
LLL                      0000000000    GGGGGGGGGGGG    IIIIIIIII    NNN    NNN
LLL                      000          000    GGG          III          NNN    NNN
LLL                      000          000    GGG          III          NNN    NNN
LLL                      000          000    GGG          III          NNN    NNN
LLL                      000          000    GGG          III          NNNNNN    NNN
LLL                      000          000    GGG          III          NNNNNN    NNN
LLL                      000          000    GGG          III          NNNNNN    NNN
LLL                      000          000    GGG          III          NNN    NNN    NNN
LLL                      000          000    GGG          III          NNN    NNN    NNN
LLL                      000          000    GGG          III          NNN    NNN    NNN
LLL                      000          000    GGG          III          NNN          NNNNNN
LLL                      000          000    GGG          III          NNN          NNNNNN
LLL                      000          000    GGG          III          NNN          NNNNNN
LLL                      000          000    GGG          III          NNN          NNN
LLL                      000          000    GGG          III          NNN          NNN
LLL                      000          000    GGG          III          NNN          NNN
LLL                      000          000    GGG          III          NNN          NNN
LLLLLLLLLLLLLLLLLLLL    0000000000    GGGGGGGGGG    IIIIIIIII    NNN    NNN
LLLLLLLLLLLLLLLLLLLL    0000000000    GGGGGGGGGG    IIIIIIIII    NNN    NNN
LLLLLLLLLLLLLLLLLLLL    0000000000    GGGGGGGGGG    IIIIIIIII    NNN    NNN

```

```
BBBBBBBBB  RRRRRRRR  EEEEEEEEE  AAAAAA  KK  KK  IIIIII  NN  NN
BBBBBBBBB  RRRRRRRR  EEEEEEEEE  AAAAAA  KK  KK  IIIIII  NN  NN
BB  BB  RR  RR  EE  AA  AA  KK  KK  II  NN  NN
BB  BB  RR  RR  EE  AA  AA  KK  KK  II  NN  NN
BB  BB  RR  RR  EE  AA  AA  KK  KK  II  NN  NN
BBBBBBBBB  RRRRRRRR  EEEEEEEEE  AA  AA  KKKKKK  II  NNNN  NN
BBBBBBBBB  RRRRRRRR  EEEEEEEEE  AA  AA  KKKKKK  II  NNNN  NN
BB  BB  RR  RR  EE  AAAAAAAAAA  KK  KK  II  NN  NN
BB  BB  RR  RR  EE  AAAAAAAAAA  KK  KK  II  NN  NN
BB  BB  RR  RR  EE  AA  AA  KK  KK  II  NN  NNNN
BBBBBBBBB  RR  RR  EEEEEEEEE  AA  AA  KK  KK  IIIIII  NN  NN
BBBBBBBBB  RR  RR  EEEEEEEEE  AA  AA  KK  KK  IIIIII  NN  NN
```

```
LL  IIIIII  SSSSSSSS
LL  IIIIII  SSSSSSSS
LL  II  SS
LL  II  SS
LL  II  SS
LL  II  SSSSSS
LL  II  SSSSSS
LL  II  SS
LL  II  SS
LL  II  SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS
```



```
1 0001 0 MODULE breakin (IDENT = 'V04-000',
2 0002 0 ADDRESSING_MODE(EXTERNAL = GENERAL)) =
3 0003 1 BEGIN
4 0004 1
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1 ++
30 0030 1 FACILITY: Login
31 0031 1
32 0032 1 ABSTRACT:
33 0033 1
34 0034 1 This module contains all the routines to scan and manipulate the
35 0035 1 Compound Intrusion Analysis blocks, add new entries to the CIA
36 0036 1 queues, locate intruders, and remove suspects.
37 0037 1
38 0038 1 ENVIRONMENT:
39 0039 1
40 0040 1 VAX/VMS operating system.
41 0041 1
42 0042 1 AUTHOR: Gerry Smith 12-July-1983
43 0043 1
44 0044 1 Modified by:
45 0045 1
46 0046 1 V03-006 ACG0436 Andrew C. Goldstein, 23-Jul-1984 17:25
47 0047 1 Add support for LGI_BRK_TERM, put result in global cell
48 0048 1
49 0049 1 V03-005 MHB0131 Mark Bramhall 5-Apr-1984
50 0050 1 Change to new TERMINAL_DEVICE flag.
51 0051 1
52 0052 1 V03-004 ACG0390 Andrew C. Goldstein, 18-Jan-1984 11:33
53 0053 1 Fix arg list mismatches (remove unused username)
54 0054 1
55 0055 1 V03-003 ACG0376 Andrew C. Goldstein, 22-Nov-1983 11:02
56 0056 1 Redesign match algorithm to reduce service denial
57 0057 1 problems, fix expiration of old entries.
```

```
58 0058 1 |
59 0059 1 | V03-002 GAS0189 Gerry Smith 22-Sep-1983
60 0060 1 | If a terminal is specified, see if there's another device,
61 0061 1 | the actual physical device, associated with the terminal.
62 0062 1 |
63 0063 1 | V03-001 GAS0185 Gerry Smith 16-Sep-1983
64 0064 1 | Add randomness to the hide time, so that the actual hide time
65 0065 1 | is between 50% and 150% of that specified by the SYSGEN
66 0066 1 | parameter.
67 0067 1 |
68 0068 1 | --
69 0069 1 |
70 0070 1 |
71 0071 1 | Include files
72 0072 1 |
73 0073 1 | LIBRARY 'SYS$LIBRARY:LIB'; ! VAX/VMS system definitions
74 0074 1 |
75 0075 1 |
76 0076 1 | Declare the linkages to allocate and deallocate nonpaged pool, as well
77 0077 1 | as grab the CIA mutex.
78 0078 1 |
79 0079 1 | LINKAGE
80 0080 1 | CVTDEV = JSB (REGISTER = 0, ! Length of output buffer,
81 0081 1 | REGISTER = 1, ! Address of output buffer
82 0082 1 | REGISTER = 4, ! Format of device name
83 0083 1 | REGISTER = 5, ! Address of UCB
84 0084 1 | REGISTER = 1), ! Length of final name
85 0085 1 | ALLO = JSB (REGISTER = 1, ! R1 = size (on input)
86 0086 1 | REGISTER = 1, ! R1 = size of block
87 0087 1 | REGISTER = 2), ! R2 = address of block
88 0088 1 | NOPRESERVE (3,4,5), ! R3, R4, R5 destroyed
89 0089 1 | DEALLO = JSB (REGISTER = 0), ! R0 = address of block
90 0090 1 | NOPRESERVE (1,2,3,4,5), ! R1-R5 destroyed
91 0091 1 | LOCK = JSB (REGISTER = 0, ! R0 = address of mutex
92 0092 1 | REGISTER = 4), ! R4 = PCB address
93 0093 1 |
94 0094 1 |
95 0095 1 |
96 0096 1 | Macro to set the processor interrupt priority level register
97 0097 1 |
98 0098 1 | BUILTIN
99 0099 1 | MTPR;
100 0100 1 |
101 0101 1 | MACRO ! set processor IPL
102 0102 1 | SET_IPL (LEVEL) = MTPR (%REF (LEVEL), PR$_IPL)%;
103 0103 1 |
```



```
: 105      0104 1 |
: 106      0105 1 | Table of contents
: 107      0106 1 |
: 108      0107 1 | FORWARD ROUTINE
: 109      0108 1 |   cia_scan,           | Set up for elevated-IPL scans
: 110      0109 1 |   check_intruder,    | Check intruder block
: 111      0110 1 |   check_suspect,     | Check suspect block
: 112      0111 1 |   add_suspect;       | Add a suspect block
: 113      0112 1 |
: 114      0113 1 |
: 115      0114 1 | External routines
: 116      0115 1 |
: 117      0116 1 | EXTERNAL ROUTINE
: 118      0117 1 |   ioc$cv_t_devnam : CVTDEV, | Construct device name from UCB
: 119      0118 1 |   sch$lockw : LOCK,        | Lock CIA mutex
: 120      0119 1 |   sch$unlock : LOCK,       | Unlock CIA mutex
: 121      0120 1 |   exe$alononpaged : ALLO,   | Allocate non-paged pool
: 122      0121 1 |   exe$deanonpaged : DEALLO; | Deallocate non-paged pool
: 123      0122 1 |
: 124      0123 1 |
: 125      0124 1 | Declare system areas
: 126      0125 1 |
: 127      0126 1 | EXTERNAL
: 128      0127 1 |   break_attempt : BYTE,
: 129      0128 1 |   terminal_device : BYTE,
: 130      0129 1 |   phy_term_name : VECTOR,
: 131      0130 1 |   uaf_record : REF $BBLOCK,
: 132      0131 1 |   org_username,
: 133      0132 1 |   ctl$gl_pcb : REF $BBLOCK,
: 134      0133 1 |   ctl$t_nodename : VECTOR[,BYTE],
: 135      0134 1 |   ctl$t_remoteid : VECTOR[,BYTE],
: 136      0135 1 |   cia$gl_mutex,
: 137      0136 1 |   cia$gl_intruder : $BBLOCK,
: 138      0137 1 |   sys$gl_brk_lim : BYTE,
: 139      0138 1 |   sys$gl_brk_tmo,
: 140      0139 1 |   sys$gl_hid_tim,
: 141      0140 1 |   exe$gl_dynamic_flags : BITVECTOR;
: 142      0141 1 |
: 143      0142 1 | EXTERNAL LITERAL
: 144      0143 1 |   exe$v_brk_term : UNSIGNED (6);
```

```
146 0144 1 GLOBAL ROUTINE cia_scan (intruder) =
147 0145 2 BEGIN
148 0146 2
149 0147 2 !+++
150 0148 2
151 0149 2 This is the routine that is called to start either a suspect
152 0150 2 or intruder scan.
153 0151 2
154 0152 2 Inputs:
155 0153 2     intruder - indicator for which type of scan to perform
156 0154 2         1 ==> intruder list
157 0155 2         0 ==> suspect list
158 0156 2
159 0157 2 Outputs:
160 0158 2     None.
161 0159 2
162 0160 2 Return status: (low bit)
163 0161 2     0 - intruder (either new, or in evasive action)
164 0162 2     1 - just another suspect, or an expired intruder
165 0163 2
166 0164 2 !---
167 0165 2
168 0166 2 BUILTIN
169 0167 2     emul,
170 0168 2     addm,
171 0169 2     cmpm,
172 0170 2     remque;
173 0171 2
174 0172 2 LABEL
175 0173 2     cia_search,
176 0174 2     cia_check;
177 0175 2
178 0176 2 LOCAL
179 0177 2     type : BYTE,
180 0178 2     data : VECTOR [cia$$_data, BYTE],
181 0179 2     time : VECTOR[2],
182 0180 2     term_length,
183 0181 2     delta,
184 0182 2     dummy,
185 0183 2     status,
186 0184 2     next,
187 0185 2     cia : REF $BLOCK;
188 0186 2
189 0187 2 !
190 0188 2     Get the current time.
191 0189 2
192 0190 2 $gettim (timadr = time);
193 0191 2
194 0192 2 !
195 0193 2     Get the data string to match on, which is meant to correspond to the
196 0194 2     source of the login. It is, in order of preference:
197 0195 2         node name and remote ID
198 0196 2         terminal if a real terminal plus username if real user
199 0197 2         original username (presumably creator of this process)
200 0198 2
201 0199 2 IF .ctl$t_nodename[0] NEQ 0                ! If node present,
202 0200 2 OR .ctl$t_remoteid[0] NEQ 0
```



```
203 0201 2 THEN                                ! get the remote node ID,
204 0202 BEGIN                                ! Copy the nodename first,
205 0203 type = cia$k_network;                ! then the remote id
206 0204 CH$COPY(.ctl$t_nodename[0],
207 0205     .ctl$t_nodename[1],
208 0206     .ctl$t_remoteid[0],
209 0207     .ctl$t_remoteid[1],
210 0208     cia$$_data,
211 0209     data);
212 0210
213 0211 END
214 0212
215 0213 ELSE IF .terminal_device
216 0214 THEN
217 0215 BEGIN
218 0216 IF .exe$gl_dynamic_flags[exe$v_brk_term]
219 0217 THEN term_length = .phy_term_name[0]
220 0218 ELSE term_length = 0;
221 0219
222 0220 IF .uaf_record NEQ 0
223 0221 THEN
224 0222 BEGIN
225 0223 type = cia$k_term_user;
226 0224 CH$COPY(.term_length,
227 0225     .phy_term_name[1],
228 0226     uaf$$_username,
229 0227     uaf_record[uaf$t_username],
230 0228     cia$$_data,
231 0229     data);
232 0230
233 0231 END
234 0232 ELSE
235 0233 BEGIN
236 0234 type = cia$k_terminal;
237 0235 CH$COPY(.term_length,
238 0236     .phy_term_name[1],
239 0237     cia$$_data,
240 0238     data);
241 0239
242 0240 END;
243 0241
244 0242 ELSE
245 0243 BEGIN
246 0244 type = cia$k_username;
247 0245 CH$COPY(uaf$$_username,
248 0246     org_username,
249 0247     cia$$_data,
250 0248     data);
251 0249
252 0250 END;
253 0251
254 0252 !
255 0253 ! Set up pointers to the address of the data. Also set STATUS = 1, to show
256 0254 ! that nothing so far.
257 0255
258 0256
259 0257 2 status = 1;
```

```
260 0258 2
261 0259 2
262 0260 2 Scan the lists, if there are any blocks.
263 0261 2
264 0262 2 cia search: BEGIN
265 0263 2 sch$lockw(cia$gl_mutex, .ctl$gl_pcb);
266 0264 2 cia = .cia$gq_intruder;
267 0265 2
268 0266 2 UNTIL .cia EQL cia$gq_intruder
269 0267 2 DO
270 0268 2 BEGIN
271 0269 2 cia_check: BEGIN
272 0270 2 next = .cia[cia$l_flink];
273 0271 2
274 0272 2 Check expiration on each entry encountered. Expiration is done with
275 0273 2 a simple compare of the entry's time against current time.
276 0274 2
277 0275 2
278 0276 2 IF cmpm (2, cia[cia$q_time], time) LEQ 0
279 0277 2 THEN
280 0278 2 BEGIN
281 0279 2
282 0280 2
283 0281 2 An expired intruder is turned back into a suspect with the
284 0282 2 count set to one short of the limit. The time is computed
285 0283 2 from the current time plus the count times the break timeout.
286 0284 2
287 0285 2
288 0286 2 IF .cia[cia$v_intruder]
289 0287 2 THEN
290 0288 2 BEGIN
291 0289 2 cia[cia$v_intruder] = 0;
292 0290 2 cia[cia$w_count] = .sys$gb_brk_lim;
293 0291 2 delta = .sys$gb_brk_lim * .sys$gl_brk_tmo;
294 0292 2 emul(%REF(10000000), ! Convert seconds into delta time
295 0293 2 delta,
296 0294 2 %REF(0),
297 0295 2 cia[cia$q_time]);
298 0296 2 addm(2, time, cia[cia$q_time], cia[cia$q_time]);
299 0297 2 END
300 0298 2
301 0299 2
302 0300 2 An expired suspect entry is removed from the list.
303 0301 2
304 0302 2
305 0303 2 ELSE
306 0304 2 BEGIN
307 0305 2 remque(.cia, dummy); ! Remove the block
308 0306 2 exe$deanonpaged(.cia); ! And return to pool
309 0307 2 LEAVE cia_check; ! No entry to check
310 0308 2 END;
311 0309 2 END;
312 0310 2
313 0311 2
314 0312 2 Now check for a match on the type we are looking for.
315 0313 2
316 0314 2
```



```
317 0315 5 IF .type EQL .cia[cia$b_subtype]
318 0316 5 AND CH$EQL (cia$s_data, data, cia$s_data, cia[cia$t_data])
319 0317 5
320 0318 5 If we have a matching block, make the appropriate checks.
321 0319 5
322 0320 5 THEN
323 0321 6 BEGIN
324 0322 6 IF .intruder
325 0323 6 THEN status = check_intruder(.cia, time)
326 0324 6 ELSE status = check_suspect(.cia, time);
327 0325 6 LEAVE cia_search;
328 0326 6 END;
329 0327 4 ! end of block CIA_CHECK
330 0328 4 cia = .next;
331 0329 4 END;
332 0330 4
333 0331 4
334 0332 4 If this is a failed login and no list entry was found, create a
335 0333 4 suspect entry.
336 0334 4
337 0335 4 IF NOT .intruder
338 0336 4 THEN add_suspect(.type, data, time);
339 0337 4 END;
340 0338 4 ! end of block CIA_SEARCH
341 0339 4
342 0340 4 Unlock the mutex and lower IPL.
343 0341 4
344 0342 4 sch$unlock(cia$gl_mutex, .ctl$gl_pcb);
345 0343 4 SET_IPL(0);
346 0344 4
347 0345 4 break_attempt = NOT .status;
348 0346 4 RETURN .status;
349 0347 1 END;
```

.TITLE BREAKIN
.IDENT \V04-000\

.EXTRN IOC\$CVT_DEVNAM, SCH\$LOCKW
.EXTRN SCH\$UNLOCK, EXE\$ALONONPAGED
.EXTRN EXE\$DEANONPAGED
.EXTRN BREAK_ATTEMPT, TERMINAL_DEVICE
.EXTRN PHY_TERM_NAME, UAF_RECORD
.EXTRN ORG_USERNAME, CTL\$GL_PCB
.EXTRN CTL\$T_NODENAME, CTL\$T_REMOTEID
.EXTRN CIA\$GL_MUTEX, CIA\$GL_INTRUDER
.EXTRN SYSS\$GB_BRK_LIM, SYSS\$GL_BRK_TMO
.EXTRN SYSS\$GL_HID_TIM, EXE\$GL_DYNAMIC_FLAGS
.EXTRN EXE\$V_BRK_TERM, SYSS\$GETTIM

.PSECT \$CODE\$,NOWRT,2

.ENTRY CIA_SCAN, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-; 0144
R11
MOVAB -64(SP), SP
PUSHL SP
CALLS #1, SYSS\$GETTIM 0190

OFFC 00000
5E C0 AE 9E 00002
00000000G 00 5E DD 00006
01 FB 00008

57	20	00000000G	00	58	00000000G	00	9A	0000F	MOVZBL	CTLST_NODENAME, R8	0199
			08			08	12	00016	BNEQ	1\$	
					00000000G	00	95	00018	TSTB	CTLST_REMOTEID	0200
			2F			13	0001E	BEQL	2\$		
		5A	03			90	00020	1\$:	MOVB	#3, TYPE	0203
		59	00		00000000G	9A	00023		MOVZBL	CTLST_REMOTEID, R9	0206
		57	38			D0	0002A		MOVL	#56, R7	0207
		56	08			9E	0002D		MOVAB	DATA, R6	
		00				2C	00031		MOVC5	R8, CTLST_NODENAME+1, #32, R7, (R6)	
			66				0003A				
			77			18	0003B		BGEQ	7\$	
		56				C0	0003D		ADDL2	R8, R6	
57	20	00000000G	00			C2	00040		SUBL2	R8, R7	
			59			2C	00043		MOVC5	R9, CTLST_REMOTEID+1, #32, R7, (R6)	
			66				0004C				
			65			11	0004D		BRB	7\$	0199
		50	00		00000000G	E9	0004F	2\$:	BLBC	TERMINAL_DEVICE, 6\$	0213
		00	00G			E1	00056		BBC	S*EXESV BRK TERM, EXESGL_DYNAMIC_FLAGS, 3\$	0216
		59	00		00000000G	D0	0005E		MOVL	PHY_TERM_NAME, TERM_LENGTH	0217
			02			11	00065		BRB	4\$	
			59			D4	00067	3\$:	CLRL	TERM_LENGTH	0218
		50	00		00000000G	D0	00069	4\$:	MOVL	PHY_TERM_NAME+4, R0	0225
		56	00		00000000G	D0	00070		MOVL	UAF_RECORD, R6	0220
			21			13	00077		BEQL	5\$	
		5A	02			90	00079		MOVB	#2, TYPE	0223
		58	38			D0	0007C		MOVL	#56, R8	0227
		57	08			9E	0007F		MOVAB	DATA, R7	
58	20		60			2C	00083		MOVC5	TERM_LENGTH, (R0), #32, R8, (R7)	
			67				00088				
			29			18	00089		BGEQ	7\$	
		57				C0	0008B		ADDL2	TERM_LENGTH, R7	
58	20	04	A6			C2	0008E		SUBL2	TERM_LENGTH, R8	
			20			2C	00091		MOVC5	#32, -4(R6), #32, R8, (R7)	
			67				00097				
			1A			11	00098		BRB	7\$	0220
		5A	01			90	0009A	5\$:	MOVB	#1, TYPE	0234
38	20		60			2C	0009D		MOVC5	TERM_LENGTH, (R0), #32, #56, DATA	0235
			08			AE	000A2				
			0E			11	000A4		BRB	7\$	0213
		5A	04			90	000A6	6\$:	MOVB	#4, TYPE	0245
38	20	00000000G	00			2C	000A9		MOVC5	#32, ORG_USERNAME, #32, #56, DATA	0246
			08			AE	000B2				
		59	01			D0	000B4	7\$:	MOVL	#1, STATUS	0257
		50	00		00000000G	9E	000B7		MOVAB	CIA\$GL_MUTEX, R0	0263
		54	00		00000000G	D0	000BE		MOVL	CTL\$GL_PCB, R4	
			00		00000000G	16	000C5		JSB	SCH\$LOCKW	
		57	00		00000000G	D0	000CB		MOVL	CIA\$GQ_INTRUDER, CIA	0264
		50	00		00000000G	9E	000D2	8\$:	MOVAB	CIA\$GQ_INTRUDER, R0	0266
		50				D1	000D9		CMPL	CIA, R0	
			03			12	000DC		BNEQ	9\$	
			008E			31	000DE		BRW	18\$	
		56	67			D0	000E1	9\$:	MOVL	(CIA), NEXT	0270
		50	01			CE	000E4		MNEGL	#1, R0	0276
04		AE	A7			D1	000E7		CMPL	20(CIA), TIME	
			0E			19	000EC		BLSS	12\$	
			08			14	000EE		BGTR	10\$	
		6E	A7			D1	000F0		CMPL	16(CIA), TIME	

10	A7	58 00	0C	2C	0C	04	13	000F4	BEQL	11\$	
				A7		04	1F	000F6	BLSSU	12\$	
				50	00000000G	50	D6	000F8	INCL	R0	
				50		50	D6	000FA	INCL	R0	
				58	00989680	50	D5	000FC	TSTL	R0	
				10		3E	14	000FE	BGTR	14\$	
				14		A7	E9	00100	BLBC	12(CIA), 13\$	0286
						01	8A	00104	BICB2	#1, 12(CIA)	0289
						00	9A	00108	MOVZBL	SY\$GB BRK LIM, R0	0290
						50	B0	0010F	MOVW	R0, 14(CIA)	
						00	C5	00113	MULL3	SY\$GL BRK TMO, R0, DELTA	0291
						8F	7A	0011B	EMUL	#10000000, DELTA, #0, 16(CIA)	0295
						6E	C0	00125	ADDL2	TIME, 16(CIA)	0296
						AE	D8	00129	ADWC	TIME, 20(CIA)	
						0E	11	0012E	BRB	14\$	0286
						67	0F	00130	REMQUE	(CIA), DUMMY	0305
						57	D0	00133	MOVL	CIA, R0	0306
						00	16	00136	JSB	EXE\$DEANONPAGED	
						2B	11	0013C	BRB	17\$	0307
						5A	91	0013E	CMPB	TYPE, 11(CIA)	0315
						25	12	00142	BNEQ	17\$	
						38	29	00144	CMPC3	#56, DATA, 24(CIA)	0316
						1D	12	0014A	BNEQ	17\$	
						0B	04	0014C	BLBC	INTRUDER, 15\$	0322
						8F	BB	00150	PUSHR	#^M<R7, SP>	0323
						02	FB	00154	CALLS	#2, CHECK_INTRUDER	
						09	11	00159	BRB	16\$	
						8F	BB	0015B	PUSHR	#^M<R7, SP>	0324
						02	FB	0015F	CALLS	#2, CHECK_SUSPECT	
						50	D0	00164	MOVL	R0, STATUS	
						17	11	00167	BRB	19\$	0325
						56	D0	00169	MOVL	NEXT, CIA	0328
						FF63	31	0016C	BRW	8\$	0266
						AC	E8	0016F	BLBS	INTRUDER, 19\$	0335
						5E	DD	00173	PUSHL	SP	0336
						AE	9F	00175	PUSHAB	DATA	
						5A	9A	00178	MOVZBL	TYPE, -(SP)	
						03	FB	0017B	CALLS	#3, ADD_SUSPECT	
						00	9E	00180	MOVAB	CIA\$GL_MUTEX, R0	0342
						00	D0	00187	MOVL	CTL\$GL_PCB, R4	
						00	16	0018E	JSB	SCH\$UNLOCK	
						00	DA	00194	MTPR	#0, #18	0343
						59	92	00197	MCOMB	STATUS, BREAK_ATTEMPT	0345
						59	D0	0019E	MOVL	STATUS, R0	0346
						04	001A1	RET			0347

; Routine Size: 418 bytes, Routine Base: \$CODE\$ + 0000

```
351 0348 1 ROUTINE check_suspect (cia, time) =
352 0349 2 BEGIN
353 0350
354 0351 |+++
355 0352
356 0353 | Check if enough login failures have occurred to change the suspect
357 0354 | to an intruder.
358 0355
359 0356 | Inputs:
360 0357 |     cia - address of the Compound Intrusion Analysis block
361 0358 |     time - address of current time of login failure
362 0359
363 0360 | Outputs:
364 0361 |     None.
365 0362
366 0363 | Return status:
367 0364 |     0 - this is an intruder
368 0365 |     1 - just another suspect
369 0366
370 0367 |---
371 0368
372 0369 BUILTIN
373 0370     emul,
374 0371     addm;
375 0372
376 0373 MAP
377 0374     time : REF VECTOR[WORD],
378 0375     cia : REF $BLOCK;
379 0376
380 0377 LOCAL
381 0378     delta : VECTOR [2];
382 0379
383 0380 |
384 0381 | Bump the count of login (perhaps breakin) attempts. Also bump the
385 0382 | expiration time of the entry.
386 0383
387 0384 | cia[cia$w_count] = .cia[cia$w_count] + 1;
388 0385 | emul (sys$gl_brk_tmo, | Convert timeout into
389 0386 |     %REF (T0000000), | delta time value
390 0387 |     %REF (0),
391 0388 |     delta);
392 0389 | addm (2, delta, cia[cia$q_time], cia[cia$q_time]);
393 0390
394 0391 |
395 0392 | If the number of attempts is greater than the number of retries allowed,
396 0393 | then remove the CIA block from the suspect queue and put it on the
397 0394 | intruder queue.
398 0395
399 0396 | IF .cia[cia$w_count] GTR .sys$gb_brk_lim
400 0397 | THEN
401 0398 |     BEGIN
402 0399 |         LOCAL
403 0400 |             semi_second;
404 0401
405 0402 |
406 0403 | Obtain a number between 10000000 and 15000000, which represents a unit of
407 0404 | time between 1.0 and 1.5 seconds.
```



```

: 408      0405      3 |
: 409      0406      3 | This is done by taking the system time, and using the longword in the
: 410      0407      3 | "middle", swapping the words of that longword, scaling that quantity
: 411      0408      3 | down to be in the range of 5000000 or less, and adding 10000000.
: 412      0409      3 |
: 413      0410      4 | BEGIN
: 414      0411      4 | MAP semi_second : VECTOR[2,WORD];
: 415      0412      4 | semi_second[0] = .time[2];           | Get middle longword,
: 416      0413      4 | semi_second[1] = .time[1];           | swapping words on the way.
: 417      0414      4 | END;
: 418      0415      4 | semi_second = .semi_second / ((1^31-1)/5000000) + 10000000;
: 419      0416      4 |
: 420      0417      4 | emul(sys$gl_hid_tim,                 | Multiply the hide time
: 421      0418      4 |      semi_second,                   | to get number of seconds
: 422      0419      4 |      %REF(0),                       |
: 423      0420      4 |      cia[cia$q_time]);              | and put it in the block
: 424      0421      4 | addm(2,                             | then add
: 425      0422      4 |      .time,                         | the current time
: 426      0423      4 |      cia[cia$q_time],               |
: 427      0424      4 |      cia[cia$q_time]);              | and put it in the block
: 428      0425      4 | cia[cia$intruder] = 1;              | Show it's an intruder
: 429      0426      4 | RETURN 0;
: 430      0427      4 | END;
: 431      0428      2 |
: 432      0429      2 | RETURN 1;
: 433      0430      1 | END;
```

```

                                000C 00000 CHECK_SUSPECT:
                                .WORD      Save R2,R3
                                5E          0C C2 00002      SUBL2      #12, SP
                                50          AC D0 00005      MOVL       CIA, R0
                                04          AE A0 B6 00009      INCW       14(R0)
                                00 00989680 8F 00000000G      EMUL      SYSSGL_BRK_TMO, #10000000, #0, DELTA
                                53          10 A0 9E 0001A      MOVAB     16(R0), R3
                                63          04 AE C0 0001E      ADDL2     DELTA, (R3)
                                04          AE D8 00022      ADWC      DELTA, 4(R3)
                                51 00000000G      00 9A 00027      MOVZBL   SYSSGB_BRK_LIM, R1
                                0E          51 B1 0002E      CMPW      R1, 14(R0)
                                51          33 1E 00032      BGEQU     1$
                                6E          AC D0 00034      MOVL      TIME, R1
                                02          A1 B0 00038      MOVW      4(R1), SEMI_SECOND
                                AE          A1 B0 0003C      MOVW      2(R1), SEMI_SECOND+2
                                52          6E 000001AD 8F C7 00041      DIVL3     #429, SEMI_SECOND, R2
                                6E 00989680 E2 9E 00049      MOVAB     10000000(R2), SEMI_SECOND
                                63 00000000G      00 7A 00050      EMUL      SYSSGL_HID_TIM, SEMI_SECOND, #0, (R3)
                                04          61 C0 00059      ADDL2     (R1), (R3)
                                0C          A1 D8 0005C      ADWC      4(R1), 4(R3)
                                50          01 88 00061      BISB2     #1, 12(R0)
                                04          04 11 00065      BRB       2$
                                50          01 D0 00067 1$:      MOVL      #1, R0
                                04 0006A      RET
                                50          D4 0006B 2$:      CLRL      R0
                                04 0006D      RET
                                04 0006D
```

BREAKIN
V04-000

M 10
16-Sep-1984 01:49:34
14-Sep-1984 12:41:04

VAX-11 BLISS-32 V4.0-742
[LOGIN.SRC]BREAKIN.B32;1

Page 12
(4)

; Routine Size: 110 bytes, Routine Base: \$CODE\$ + 01A2


```

: 435      0431 1 ROUTINE check_intruder (cia, time) =
: 436      0432 2 BEGIN
: 437      0433 2
: 438      0434 2 |+++
: 439      0435 2
: 440      0436 2 Check if list entry is an intruder (for otherwise successful
: 441      0437 2 validation).
: 442      0438 2
: 443      0439 2 Inputs:
: 444      0440 2     cia - address of intruder block
: 445      0441 2     time - current time
: 446      0442 2
: 447      0443 2 Outputs:
: 448      0444 2     none.
: 449      0445 2
: 450      0446 2 Return status:
: 451      0447 2     0 - this is an intruder, perform evasive action
: 452      0448 2     1 - this is only suspect, no action
: 453      0449 2
: 454      0450 2 |---
: 455      0451 2
: 456      0452 2 MAP
: 457      0453 2     cia : REF $BBLOCK;
: 458      0454 2
: 459      0455 2 LOCAL
: 460      0456 2     dummy;
: 461      0457 2
: 462      0458 2 |
: 463      0459 2 | If this is not an intruder entry, take no action.
: 464      0460 2
: 465      0461 2 IF NOT .cia[cia$w_intruder]
: 466      0462 2 THEN RETURN 1;
: 467      0463 2
: 468      0464 2 cia[cia$w_count] = .cia[cia$w_count] + 1;
: 469      0465 2 RETURN 0;
: 470      0466 1 END;
```

0000 00000 CHECK_INTRUDER:						
50	04	AC	D0 00002	.WORD	Save nothing	: 0431
04	0C	A0	E8 00006	MOVL	CIA, R0	: 0461
50		01	D0 0000A	BLBS	12(R0), 1\$	
			04 0000D	MOVL	#1, R0	: 0462
	0E	A0	B6 0000E 1\$:	RET		
		50	D4 00011	INCW	14(R0)	: 0464
			04 00013	CLRL	R0	: 0465
				RET		: 0466

; Routine Size: 20 bytes, Routine Base: \$CODE\$ + 0210

```

472 0467 1 ROUTINE add_suspect (type, data, time) =
473 0468 2 BEGIN
474 0469 2
475 0470 2 |+++
476 0471 2 |
477 0472 2 |   Add a suspect block
478 0473 2 |
479 0474 2 |   Inputs:
480 0475 2 |       type - type code of data
481 0476 2 |       data - data string to match on
482 0477 2 |       time - time of login attempt
483 0478 2 |
484 0479 2 |   Outputs:
485 0480 2 |       None.
486 0481 2 |
487 0482 2 |---
488 0483 2
489 0484 2 BUILTIN
490 0485 2     emul,
491 0486 2     addm,
492 0487 2     insque;
493 0488 2
494 0489 2 LOCAL
495 0490 2     size,
496 0491 2     cia : REF $BBLOCK;
497 0492 2
498 0493 2 |
499 0494 2 |   Get a chunk of non-paged pool.
500 0495 2 |
501 0496 2 |   IF NOT exe$alononpaged(cia$c_length; size, cia)
502 0497 2 |   THEN RETURN 1;
503 0498 2 |
504 0499 2 |
505 0500 2 |   Fill in the block.
506 0501 2 |
507 0502 2 |   cia[cia$w_size] = .size;
508 0503 2 |   cia[cia$b_type] = dyn$c_cia;
509 0504 2 |   cia[cia$b_subtype] = .type;
510 0505 2 |   cia[cia$w_flags] = 0;
511 0506 2 |   cia[cia$w_count] = 1;
512 0507 2 |   emul (sys$gl_brk_tmo,
513 0508 2 |       %REF (T0000000),
514 0509 2 |       %REF (0),
515 0510 2 |       cia[cia$q_time]);
516 0511 2 |   addm (2, .time, cia[cia$q_time], cia[cia$q_time]);
517 0512 2 |   CH$MOVE(cia$s_data,
518 0513 2 |       .data,
519 0514 2 |       cia[cia$t_data]);
520 0515 2 |
521 0516 2 |   insque(.cia, .cia$gq_intruder[cia$l_blink]);
522 0517 2 |
523 0518 2 |   RETURN 1;
524 0519 1 END;
```

! Put in size
! Type
! Data type
! This is a suspect
! Tried one time
! Convert timeout into
! delta time
! and add to current time
! And this is the
! match data
! Put at end of intruder queue

				OFFC 00000 ADD_SUSPECT:			
		51	50	8F 9A 00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	: 0467
			00000000G	00 16 00006	MOVZBL	#80, R1	: 0496
		56		52 D0 0000C	JSB	EXE\$ALONONPAGED	
		42		50 E9 0000F	MOVL	R2, R6	
	08	A6		51 B0 00012	BLBC	R0, 1\$	
	0A	A6	45	8F 90 00016	MOVW	SIZE, 8(CIA)	: 0502
	0B	A6	04	AC 90 0001B	MOVB	#69, 10(CIA)	: 0503
	0C	A6	00010000	8F D0 00020	MOVB	TYPE, 11(CIA)	: 0504
10	A6		00 00989680	8F 00000000G	MOVL	#65536, 12(CIA)	: 0505
		50	0C	00 7A 00028	EMUL	SY\$GGL_BRK_TMO, #10000000, #0, 16(CIA)	: 0510
		10		BC 9E 00036	MOVAB	@TIME, R0	: 0511
	14	A6	04	60 C0 0003A	ADDL2	(R0), 16(CIA)	
	08	BC		A0 D8 0003E	ADWC	4(R0), 20(CIA)	
18	A6			38 28 00043	MOV3	#56, @DATA, 24(CIA)	: 0514
		50	00000000G	00 9E 00049	MOVAB	CIA\$GQ_INTRUDER+4, R0	: 0516
		00		66 0E 00050	INSQUE	(CIA), @0(R0)	
		50		01 D0 00054	MOVL	#1, R0	: 0518
				04 00057	RET		: 0519

; Routine Size: 88 bytes, Routine Base: \$CODE\$ + 0224

BREAKIN
V04-000

D 11
16-Sep-1984 01:49:34
14-Sep-1984 12:41:04

VAX-11 Bliss-32 V4.0-742
[LOGIN.SRC]BREAKIN.B32;1

Page 16
(7)

: 526
: 527
0520 1 END
0521 0 ELUDOM

PSECT SUMMARY

:
: Name Bytes Attributes
: \$CODE\$ 636 NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

:
: File ----- Symbols ----- Pages Processing
: Total Loaded Percent Mapped Time
: _\$255\$DUA28:[SYSLIB]LIB.L32;1 18619 24 0 1000 00:01.4

COMMAND QUALIFIERS

:
: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS\$:BREAKIN/OBJ=OBJ\$:BREAKIN MSRC\$:BREAKIN/UPDATE=(ENH\$:BREAKIN)

: Size: 636 code + 0 data bytes
: Run Time: 00:08.6
: Elapsed Time: 00:35.8
: Lines/CPU Min: 3626
: Lexemes/CPU-Min: 19593
: Memory Used: 132 pages
: Compilation Complete

0221 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

